# Assignment 3 Statistical Learning

Code

1.

Hide

```
library(MASS)
library(rmarkdown)
library(tinytex)
```

Hide

```
gen_data <- function(n) {
  p <- 15
  n1 <- n2 <- n/2
  cov_1 <- diag(rep(1,p)) + 0.2
  x_class1 <- mvrnorm(n1, mu = rep(3,p), Sigma = cov_1)
  x_class2 <- mvrnorm(n2, mu = rep(2,p), Sigma = cov_1)
  x <- rbind(x_class1, x_class2)
  y <- rep(c(1,2), c(n1, n2))
  df <- as.data.frame(cbind(x,y))
  names(df) <- c(paste0("x", 1:p), "y")
  return(df)
}
```

Hide

```
train_set_50 <- gen_data(50)
train_set_10000 <-gen_data(50)
test_set_10000 <-gen_data(50)
```

Hide

```
sum_lda_mean_50 <- 0

for (i in 1:100){
  set.seed(123)
  train_set_50 <- gen_data(50)
  test_set_10000 <- gen_data(10000)
  lda.fit <- lda(y ~ x1 + x2 + x3, data = train_set_50)
  lda.pred <- predict(lda.fit, test_set_10000)
  lda_mean_50 <- mean(lda.pred$class == test_set_10000$y)
  sum_lda_mean_50 <- sum_lda_mean_50 + lda_mean_50

}
avg_lda_mean_50 <- sum_lda_mean_50/100
print(avg_lda_mean_50)
```

```
[1] 0.7364
```

Hide

```
sum_lda_mean_10000 <- 0

for (i in 1:100){
  set.seed(123)
  train_set_10000 <- gen_data(10000)
  test_set_10000 <- gen_data(10000)
  lda.fit <- lda(y ~ x1 + x2 + x3, data = train_set_10000)
  lda.pred <- predict(lda.fit, test_set_10000)
  lda_mean_10000 <- mean(lda.pred$class == test_set_10000$y)
  sum_lda_mean_10000 <- sum_lda_mean_10000 + lda_mean_10000

}
avg_lda_mean_10000 <- sum_lda_mean_10000/100
print(avg_lda_mean_10000)
```

```
[1] 0.7538
```

Hide

```
library(ISLR2)
train_set_10000$y<-ifelse(train_set_10000$y==1,0,1)
log_reg <- glm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 + x12 + x13 + x14 + x15, data = train_s
et_10000, family = "binomial")
predict <- predict(log_reg, test_set_10000, type = 'response')
glm_preds[predict > .5] <- 1
mean(predict == test_set_10000$y)
```

```
[1] 0
```

Hide

```
get_TPR <- function(preds, y, pos_class = 1){
sum(y == pos_class & preds == pos_class) / sum(y == pos_class)
}
get_TNR <- function(preds, y, neg_class = 0){
sum(y == neg_class & preds == neg_class) / sum(y == neg_class)
}
get_BAC <- function(preds,y, pos_class = 1, neg_class = 0){
TPR <- get_TPR(preds,y, pos_class)
TNR <- get_TNR(preds,y, neg_class)
mean(c(TPR, TNR))
}
get_BAC(glm_preds, test$Direction, pos_class = "Up", neg_class = "Down")
```

```
[1] NaN
```

Hide

```

train_set_10000$y <- ifelse(train_set_10000$y==1, 0, 1)
glm.y <- glm(y ~ x1 + x2 + x3, data = train_set_10000, family = "binomial")
glm.predict.y <- predict(glm.y, train_set_10000, type = 'response')

#test_set_10000$predict.y <- ifelse(glm.predict.y >= 0.5, 1,0)
accuracy = mean(train_set_10000$y == test_set_10000$predict.y)
print(accuracy)
```

```
[1] 0.2463
```